

Game Proposal: Pinball Luminary

CPSC 427 - Video Game Programming

Team Members:

Chenwen Shen 13454772

Erin Zhang 54691282

Tianhong Zhou 14355010

Xueyong Zhang 25012923

Wenhan Zhu 50497973

Kevin Shen 85809796

Story:

An evil demon lord threatens to destroy the entire world and is up to our hero, the great pinball luminary, to save it. With nothing but a flipper, our hero must venture up the twisting and ever-changing maze of the demon lord's tower, defeating all of the demon's minions and finding priceless artifacts to aid their journey ahead. Will our hero make it to the top of the tower and save the world, or will they be defeated by the monsters in the tower and be lost forever?

Technical Elements:

Identify how the game satisfies the core technical requirements: rendering; geometric/sprite/other assets; 2D geometry manipulation (transformation, collisions, etc.); gameplay logic/AI, physics.

Rendering:

We will need to render each individual dungeon room as the player navigates through them. Upon interacting with an enemy, a scene change occurs and the combat scene will be rendered.

2D dynamic shadows can be casted on the character as well as game objects to make the scene more appealing and realistic.

A cartoon shader will be applied to give the game a better, textured look.

Physics:

Collision detection, acceleration calculation, gravity sim, rigid body physics for pinball combat

Hitbox creation for combat scene objects

Combat scene objects needs various physics parameters like elasticity, etc

2D geometry manipulation:

Collision effects

Advanced Technical Elements:

List the more advanced and additional technical elements you intend to include in the game prioritized on likelihood of inclusion. Describe the impact on the gameplay in the event of skipping each of the features and propose an alternative.

- procedural generation to generate a randomized dungeon each playthrough of the game
- enemy AI during the dungeon portion so that enemies follow and attack the player when the player enters a room
- a map system that displays the layout of the dungeon
- complex enemies during the pinball combat that may have complex geometry and animation or AI patterns.
- sync up battle with the music beats

Devices:

Explain which input devices you plan on supporting and how they map to in-game controls.

Support for PC with keyboard and mouse controls.

WASD controls for moving across the dungeon.

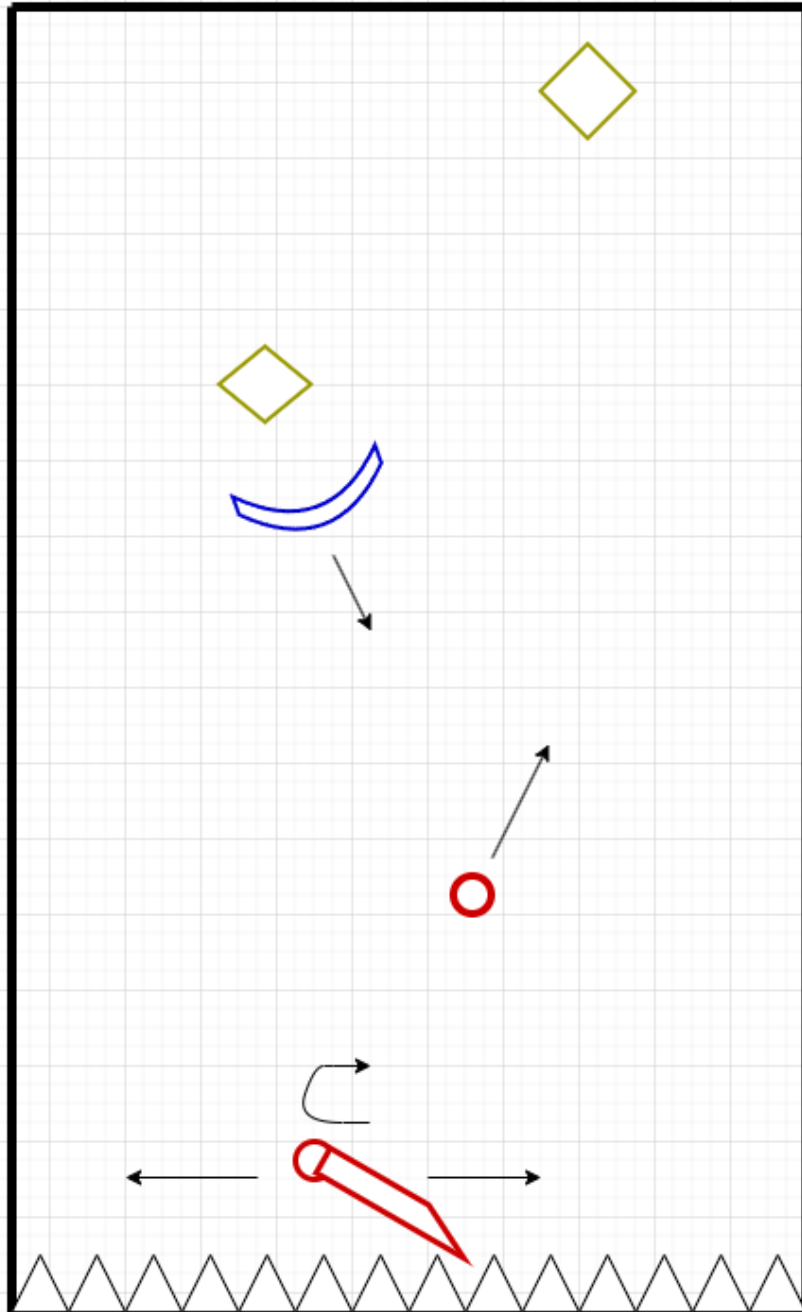
Mouse x location mapped to flipper location to move during combat

Shift to switch flipper direction & player polarity

Left click to dash/flipper activate

Concepts:

Produce basic, yet descriptive, sketches of the major game states (screens). These should be consistent with the game design elements, and help you assess the amount of work to be done.



Tools:

Specify and motivate the libraries and tools that you plan on using except for C/C++ and OpenGL.

We may use a coding library for handling text and UI elements.

We may use clip studio paint for making custom art assets and/or other similar art tools for creating sprites and assets.

We will use a free sound bank for music and sound effects.

Team management:

Identify how you will assign and track tasks, and describe the internal deadlines and policies you will use to meet the goals of each milestone.

We will use the Github issue tracker for any problems that may arise and for any tasks that need to be finished and also Discord for communication and to keep track of any documents and ideas for the game. We will have meetings when needed to discuss any issues with the game.

Development Plan:

Provide a list of tasks that your team will work on for each of the weekly deadlines. Account for some testing time and potential delays, as well as describing alternative options (plan B). Include all the major features you plan on implementing (no code).

Week: Sept 31 - **Skeletal Game**

- Layout basic class hierarchy/structure
- Basic design for pinball game
- Basic collision detection for pinball game
- Basic design for dungeon

Week: Oct 8

- Basic movement across the dungeon
- Basic scene for dungeon completed (game space boundaries, some actions for character/assets)
- Basic collision detection for dungeon (at least for boundaries)
- Basic scene change

Week: Oct 15

- Basic dungeon generation logic

- Basic dungeon progression system (cards the player chooses after combat or collect items)
- Render some art assets with game

Week: Oct 22 - **Minimal Playability**

- (Smooth) enemy collision and scene change
- Tutorial design
- Different types of dungeon rooms
- Sprite sheet animation
- Integrate more art assets

Week: Oct 29

- Implement a basic tutorial
- Basic randomized dungeon generation logic
- Improved dungeon progression system
- Scale difficulty of combat with dungeon progression

Week: Nov 5 - **Playability**

- Procedural generation for dungeon
- Advanced physics effects for pinball combat
- Integrate all art assets needed for the game
- Ensure robustness of game

Week: Nov 12

- Save/Load feature
- More advanced physics effects or complex geometry, possibly more advanced enemies
- Particle Effects

Week: Nov 19 - **Final Game**

- Audio feedback
- Full tutorial system
- User testing

Week: Nov 26

- Game balancing

Team member roles:

Tianhong

- Initial rendering

Erin

- Initial rendering
- Control of player moving through dungeon

Wendy

- basic dungeon design
- make art assets
- integrate assets with rendering

Kevin

- Enemy interaction and scene change
- Basic dungeon framework

Xueyong:

- UI library exploration

Wenhan

- Physics system
- Collision detection
- Combat design